Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Herzog, Dennis; Krüger, Volker; Grest, Daniel

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

AALBORG UNIVERSITY

Copenhagen

Computer Vision and Machine Intelligence Lab (CVMI)

Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Dennis Herzog, Volker Krüger, Daniel Grest

Advances in Computer Vision and Machine Intelligence CVMI 2008:1

ISSN 1902-2034

Dennis Herzog, Volker Krüger, Daniel Grest
Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Report number: CVMI 2008:1
ISSN 1902-2034
Publication date: January 2008
E-mail of author: vok@cvmi.aau.dk

Reports can be ordered from:

Computer Vision and Machine Intelligence Lab
Aalborg University Copenhagen (AAU)
DK-2970 Ballerup
DENMARK

telefax: +45 96 35 24 80
http://www.cvmi.aau.dk/

Parametric Hidden Markov Models for Recognition and Synthesis of Movements

Dennis Herzog, Volker Krger, Daniel Grest
Computer Vision and Machine Intelligence Lab
Aalborg University Copenhagen

January 7, 2008

Abstract

A common problem in human movement recognition is the recognition of movements of a particular type (semantic). E.g., grasping movements have a particular semantic (grasping) but the actual movements usually have very different appearances due to, e.g., different grasping directions. In this paper, we develop an exemplar-based parametric hidden Markov model (PHMM) that allows to represent, e.g., movements of a particular type and that compensates for the different appearances and parameterizations of that movement. The PHMM is based on exemplar movements that have to be ”demonstrated” to the system. Recognition and synthesis are carried out through locally linear interpolation of the exemplar movements. For a meaningful interpolation, the exemplars have to be in sync, what exhibits certain problems that are resolved in this paper. In our experiments we combine our PHMM approach with our 3D body tracker. Experiments are performed with pointing and grasping movements. Synthesis for grasping is parameterized by the positions of the objects to be grasped. In case of recognition, our approach is able to recover the position of an object at which a human volunteer is pointing. Our experiments show the flexibility of the PHMMs in terms of the amount of training data and its robustness in terms of noisy observation data. In addition, we compare our PHMM to an other kind of PHMM, which has been introduced by Wilson and Bobick.

Keywords: action recognition, action representation, computer vision, robotics, AI

1 Introduction

One of the major problems in action and movement[1] recognition is to recognize actions that are of the same type but can have very different appearances depending on the situation they appear in. In addition, for some actions these differences are of major importance in order to convey their meaning. Consider for example the movement of a human pointing at an object, “This object there...”, with the finger pointing at a particular object (like in Fig. 1). Clearly, for such an action, the action itself needs to be recognized but also the spot in 3D space at which the human is pointing. Only together do these two pieces of information convey the full semantics of the movement. Another common problem is the synthesis of action: This concerns two major problem areas: In robotics, one is interested in teaching robots through simple demonstrations (imitation learning) [1, 2, 10]. In 3D human body tracking, one is interested in using motion models in order to constrain the parameter space (e.g. [8] for simple cyclic motions). In both cases, one is interested in teaching the system in an easy and efficient manner a particular movement so that afterwards, the system is able to synthesize movements of the same type, however, with a different parameterization. Here, we consider grasping movements as an example where a human is reaching out for an object to grasp it[2]. One may perform as demonstration a set of grasping movements. All grasping

[1] We use the terms *action* and *movement* interchangingly. Actions usually denote movements that involve objects.

[2] The precise choice of a hand grasp depends on the type of object, from where it is being grasped, etc. In our discussion, we omit the issue of the different hand grasps and focus only on the arm movements.

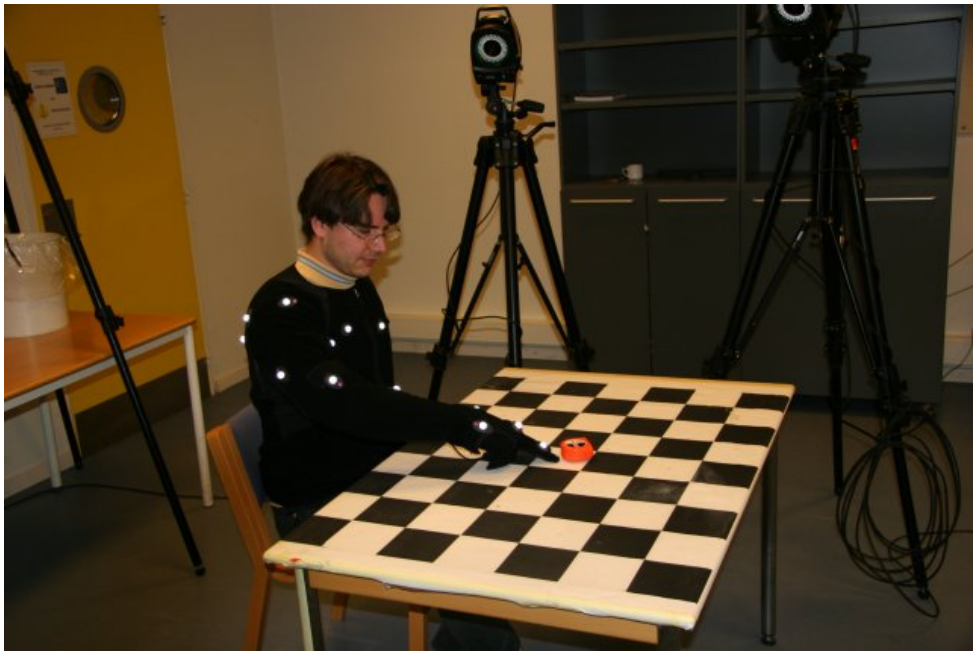

Figure 1: The image shows the setup of the capturing session for our dataset. The person is currently pointing at a raster position at the table-top.

movements depend on the location of the object to be grasped. In case of a humanoid robot, the synthesis should then allow the robot to perform the learned grasping movements with new parameterizations, e.g., grasping objects at different positions. In case of the 3D body tracking, synthesis would allow a better prediction of the next pose and even allows an estimate of parametric actions instead of the full joint configuration which would result into a considerable reduction in search space.

Most current approaches model movements with a set of movement *prototypes*, and identify a movement by identifying the prototype which explains the observed movement best. This approach, however, has its limits concerning efficiency when the space of possible parameterizations is large.

A pioneering work in this context was done by Wilson and Bobick [12]. Wilson and Bobick presented a parametric HMM approach that is able to learn an HMM based on a set of demonstrations. Their training and recognition approach is based on the EM algorithm, where the parameters of the movements are taken as latent variables. For recognition, they recover the parameters that explains best the observation.

In this paper, we develop a different parametric hidden Markov model approach. Contrary to Wilson and Bobick, our aim is recognition as well as synthesis. Also, we would like to provide a simpler and more efficient training strategy by being able to simply provide exemplars based on which the generation of novel HMMs can be done.

A further contribution is a novel method for time warping of HMM training data that is not limited to pairwise warps like the classical time warping approaches.

In the following section, we give a short overview of the related work. In Sect. 3 we provide some basics to introduce our exemplar-based parametric HMM in Sec. 4. Extensive experimental results including a comparison with [12] are presented in Sect. 5. Conclusions in Sect. 6 complete our paper.

2 Related Work

Most approaches for movement representation that are of interest in our problem context are trajectory based: Training trajectories, e.g., sequences of human body poses, are encoded in a suitable manner. Newly incoming trajectories are then compared with the previously trained ones. A recent review can be

found in [7].

Some of the most common approaches to represent movement trajectories use hidden Markov models (HMMs) [4, 9]. HMMs offer a statistical framework for representing and recognition of movements. One major advantage of HMMs is their ability to compensate for some uncertainty in time. However, due to their nature, HMMs are only able to model specific movement trajectories, but they are not able to generalize over a class of movements that vary accordingly to a specific set of parameters.

One possibility to recognize an entire class of movements is to use a set of hidden Markov models (HMMs) in a mixture-of-experts approach, as first proposed in [5]. In order to deal with a large parameter space one ends up, however, with a lot of experts and a large amount of training becomes necessary.

Another extension of the classical HMMs into parametric HMMs was presented in [12], as mentioned above. A more recent approach was presented by [1]. In this work, the interpolation is carried out in spline space where the trajectory of the end-effector is modeled. Apart from the fact that the authors have not yet performed an evaluation of their system, their approach does not seem suitable for controlling the entire arm movements for movement synthesis and recognition.

In addition to HMMs, there are also other movement representations that are interesting in our context, e.g., [6, 11]. However, these approaches share the same problems as the HMM based approaches.

3 Preliminaries of HMMs

A hidden Markov model is a probabilistic finite state machine extended in a probabilistic manner, that is defined as a triple $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, where the transition matrix $\boldsymbol{A} = (a_{ij})$ defines the transition probability between the hidden states $i, j = 1, \ldots, N$, and $\boldsymbol{B}$ defines the output distributions $b_i(\boldsymbol{x}) = \mathrm{P}(\boldsymbol{x}|q_t = i)$ of the states. The vector $\boldsymbol{\pi}$ defines the probabilities of each state of being the initial state of a hidden state sequence.

In our approach a restrictive type of continuous left-right HMMs (as in [3]) is used, whose output probability distributions of each state i are modeled by single Gaussian distribution $b_i(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, and whose state transitions are self-transitions or are transitions leading to the successor state, i.e., other transition probabilities are set to zero.

If such an HMM is used to model, e.g., a simple trajectory or sequence $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_t \ldots \boldsymbol{x}_T$, then each Gaussian $\mathcal{N}_i(\boldsymbol{x}) := b_i(\boldsymbol{x})$ would "cover" some part of the trajectory, where the state i increases as the time of the trajectory evolves. In addition, the temporal behavior of the trajectory is coded in the transition probabilities. In the case of multiple trajectories of the same kind, the Gaussian capture the variance of these trajectories, but in addition, such a model can compensate for different progression rates between the trajectories. As we want to facilitate the synthesis of movements it is obviously necessary to use left-right HMMs. However, it is worth to be mentioned that, even in the case of such a restrictive left-right model, there is no strict assignment between states and observations $\boldsymbol{x}_t$.

For a comprehensive introduction to HMMs, we refer to [4, 9]. The most important algorithms of the HMM framework are mentioned in the following example section.

3.1 Recognition using HMMs

For recognition or classification HMMs are generally used as follows: For each specific class k of sequences an HMM $\boldsymbol{\lambda}^k$ is trained by a representative training set $\mathcal{X}^k$ for that class. The training of an HMM $\boldsymbol{\lambda}$ is done by adjusting the model parameters to values, which are maximizing the likelihood function $\mathrm{P}(\mathcal{X}|\boldsymbol{\lambda})$. For this maximization, we apply the Baum/Welch expectation maximization (EM) algorithm [9].

The classification of a specific output sequence $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_T$ is done by selecting that class k, for which the likelihood $\mathrm{P}(\boldsymbol{X}|\boldsymbol{\lambda}^k)$ is maximal. The probability of a sequence $\boldsymbol{X}$ given the model is efficiently computed by the forward/backward algorithm [9].

One obvious approach for handling whole classes of parameterized actions for the purpose of action recognition and parameter estimation is a mixture-of-experts approach [5] and to sample the parameter space by training for each sample a prototype HMM. The HMM that maximizes the likelihood—given an action sequence—identifies class membership and the parameterization of the action. However, this approach is not appropriate, because too many repetitions of the action are needed to train the prototype HMMs of all samples. Therefore, we introduce the parameterization of the movements as a new parameter of the model, which also is the basic idea of the approach in [12].

4 Parametric HMM Framework

The main idea of our approach for handling whole classes of parameterized actions is a supervised learning approach where we deduce an HMM for novel action parameters by locally linear interpolation of exemplar HMMs that were previously trained on exemplar movements with known parameters. The generation of newly parameterized HMMs can be done online or offline.

The deduction of an HMMs $\boldsymbol{\lambda}^{\boldsymbol{\phi}}$ for a specific parameter is carried out by component-wise linear interpolation of the nearby exemplar models. That results, e.g., in case of a single scalar parameter u and two given exemplar models $\boldsymbol{\lambda}^0$ and $\boldsymbol{\lambda}^1$ for $u = 0, 1$, in a state-wise or Gaussian-wise deduction of the Gaussian $\mathcal{N}_i^u(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i^u, \boldsymbol{\Sigma}_i^u)$ of the state i of the model $\boldsymbol{\lambda}^u$ with means and covariances, as given by

$$\begin{aligned} \boldsymbol{\mu}_i^u &= (1-u)\boldsymbol{\mu}_i^0 + u\boldsymbol{\mu}_i^0 \\ \boldsymbol{\Sigma}_i^u &= (1-u)\boldsymbol{\Sigma}_i^1 + u\boldsymbol{\Sigma}_i^1. \end{aligned} \tag{1}$$

This situation of two exemplar models $\boldsymbol{\lambda}^0$ and $\boldsymbol{\lambda}^1$ for $u = 0, 1$ is scetched in Fig. 2 for sequences of parameterization $u = 0$, and $u = 1$. Obviously, in the case of such an arrangement, the state-wise interpolation results in a good model $\boldsymbol{\lambda}^u$ for sequences, e.g., in the middle (where $u = 0.5$). But this is the case *only if* the same two states of the exemplar HMMs do model the same semantical part of the motion. Consider, e.g., the n-th state of each of the two HMMs, where one of the two states state possibly models a part of a forward motion of a hand while the other might model a part of a backward motion. Clearly, interpolation of two such states does not make sense. Therefore, we develop an alignment of the states as described in Sec. 4.2 below. The expansion to the multi-variate case of parameterization $\boldsymbol{\phi}$ is straightforward, e.g., by using bilinear ($\boldsymbol{\phi} = (u, v)$) or trilinear interpolation.

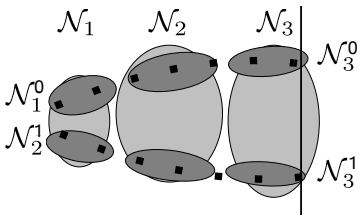

Figure 2: The upper three dark ellipsoids are depicting the Gaussians $\mathcal{N}_1^0, \ldots, \mathcal{N}_3^0$ of the states $i = 1, 2, 3$ of an HMM $\boldsymbol{\lambda}^0$ that is trained by sequences, that begin on the left and are leading to the upper part of the vertical line on the right hand side. In this case the parameterization of the sequences is $u = 0$. The dots sketch one of these training sequences. Similarly, the lower three ellipsoids of an HMM $\boldsymbol{\lambda}^1$ model sequences with a parameter $u = 1$. Additionally, the Gaussians $\mathcal{N}_i$ of a global model $\boldsymbol{\lambda}$ are indicated in light gray. In this case λ is trained with all training sequences.

4.1 Synthesis

Suppose a given grasp position $\boldsymbol{p}$ on a table. Then, synthesis can be done as follows: At first, four HMMs $\boldsymbol{\lambda}^i{}_{,i=1,\ldots,4}$ with closest associated grasp positions $\boldsymbol{p}^i$ are chosen under the constraint that at least three of the $\boldsymbol{p}^i$ are strongly not collinear and that $\boldsymbol{p}$ lies accurately in the convex hull of $\{\boldsymbol{p}^i\}$. Then, the bilinear interpolation parameters u, v are estimated such that the interpolated point $\boldsymbol{p}^{uv}$ approximates $\boldsymbol{p}$ best. Then, the model $\boldsymbol{\lambda}^{uv}$, i.e., the sequence $\boldsymbol{\mu}_1^{uv} \ldots \boldsymbol{\mu}_N^{uv}$ of the Gaussians, is calculated. Afterwards, this sequence can be expanded to a function $\boldsymbol{f}(t)$ by using spline interpolation (we use linear spline interpolation). If needed, this can be done with respect to the time durations coded in the transition probabilities.

4.2 Synchronized Setup of HMM States

As mentioned above, it is necessary to setup corresponding states of local exemplar HMMs in such a way, that the corresponding states model the same semantical parts of the movements. This task is somehow similar to dynamic time warping. The time warping algorithms synchronize sequences to compensate for different dynamics. But these algorithms are not suitable for our task. On the one hand, these algorithms synchronize sequences only pairwise. On the other hand, the alignment of sequences do not overcome the task of setting up the exemplar HMMs. Here, it is worth to mention, that we have successfully used HMMs for time warping—in a not pairwise way—of several sequences, which do vary, considerably.

Here, the underlying idea is to set up local exemplar HMMs $\boldsymbol{\lambda}^\phi$ by using the ability of HMMs to compensate to some extend temporal variations. We precede in two steps: In the first step a global HMM $\boldsymbol{\lambda}$ is trained based on the whole training set $\mathcal{X}$ that contains movements of different parameterizations ϕ, but of the same type. Such a global HMM is sketched in Fig. 2 with the light gray ellipsoids/Gaussians. The situation that movements of different parameterizations are covered in such a symmetrical way as in Fig. 2 can be enforced, in some way, by enforcing the hidden state sequences to pass the states always in the same sequential order from state 1 to state N. This is caused by the choice of the type of left-right model, and by allowing only sequences that start in the first and end in the last state.

In the second step, consider the reduced training set $\boldsymbol{\lambda}^\phi$ of a specific parameterization ϕ. On this training set we train an exemplar HMM $\boldsymbol{\lambda}^\phi$ while using the parameters of the global HMM $\boldsymbol{\lambda}$ as initial values. In the terminology of the EM algorithm, the exemplar model $\boldsymbol{\lambda}^\phi$ for $\mathcal{X}^\phi$ is computed using $\boldsymbol{\lambda}$ as an initial configuration and by fixing the means of the Gaussians after the first EM iteration. It is worth to note, that this gives the wanted result: In the first E step of EM the posterior probabilities $\gamma_t^k(i) = \mathrm{P}(q_t = i|\boldsymbol{X}^k, \boldsymbol{\lambda})$ of being in state i at time t given the global model are computed for each sequence $\boldsymbol{X}^k = \boldsymbol{x}_1 \ldots x_T$ of the training set $\mathcal{X}^\phi$. Thus, $\gamma_t^k(i)$ defines the somehow the "responsibility" of state i for generating $\boldsymbol{x}_t^k$. In the M step the mean $\boldsymbol{\mu}_i$ of the Gaussian of state i is re-estimated as an $\gamma_t^k(i)$-weighted mean:

$$\boldsymbol{\mu}_i^k = \frac{\sum_{tk} \gamma_t^k(i) \boldsymbol{x}_t^k}{\sum_{tk} \gamma_t(i)} \tag{2}$$

If one considers the case of Fig. 2 and the depicted upper sequence $\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_7$ the responsibilities $\gamma_t(i)$ would be large for $t = 1, 2$ and $i = 1$ but small for $i > 1$ (and $t = 1, 2$) caused by the position of the Gaussian of state $i = 1$. This way $\boldsymbol{\mu}_1$, as calculated by Eq. (2), lies between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, as required. One issue gives raise to problems concerning the setup of our PHMM. The Gaussians of the global HMM that is used for the alignment of the exemplar HMMs should cover the movements of exemplar movements of different parameterization in a symmetrical way as shown in Fig. 3, and described in Sec. 4.2. However, sometimes the global HMM takes a form, where some Gaussians model *only* movements of certain parameterizations—similar to the Gaussians on the right of Fig. 3. This is not surprising if one consider the ability of HMM to compensate for temporal variations, even in our restrictive left-right model. Such an HMM can be a good model for a sequence, even though one state does not fit for the sequence,

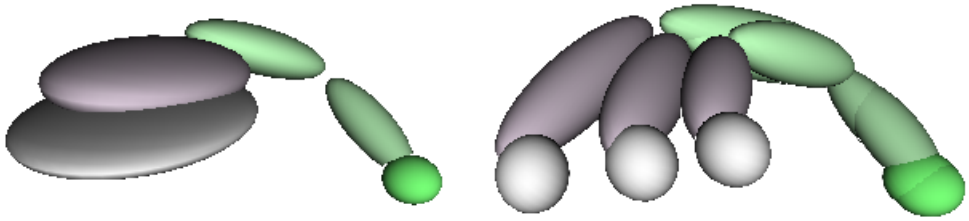

Figure 3: Synchronized setup of HMMs. Left: Some Gaussians of a global HMM are depicted on the left, which model index finger trajectories leading from the right (green ball) to the left, where the disc like ellipsoid of a Gaussian models finger positions for all pointed at positions on a table. This global HMM is used to setup the local exemplar HMMs for specific positions in a synchronized way (right).

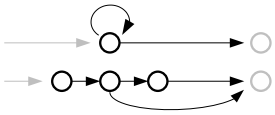

Figure 4: Time Durations of States. The upper state of a left-right HMM is replaced by the lower three pseudo states, so that the state duration lays between 2 and 3.

because, a hidden state sequence can pass a state that doesn't fit in one step and can stay for several time steps in suitable states. We addressed that problem by adding explicit time durations to the states of the HMM. For simplicity we replaced each state of the left-right HMM by some pseudo states which share one Gaussian (compare Fig. 4). This forces the hidden states sequences to stay in a state, e.g., as in Fig. 4, for at least two and for maximal three time steps.

4.3 Recognition and Parameters

In this section, we describe the recognition of the type and the parameterization of the recognized type of a parameterized movement. This is straight forward compared to the nonparametric case of classification. Consider a given sequence $\boldsymbol{X}$. We precede in two steps: First, for each possible movement type k the most likely parameter $\boldsymbol{\phi}_k$ of the corresponding parameterized HMM $\boldsymbol{\lambda}_k^{\boldsymbol{\phi}}$ is estimated. We maximize $f_k(\boldsymbol{\phi}) = P(\boldsymbol{X}|\boldsymbol{\lambda}_k^{\boldsymbol{\phi}})$ under the constraint of senseful values (e.g., $\boldsymbol{\phi} \in [0-\varepsilon, 1+\varepsilon]^d$) by using gradient descent. The next step is the recognition of the action type. Now, the classification is reduced to the classical way by choosing the most likely model $\boldsymbol{\lambda}_k = \boldsymbol{\lambda}_k^{\boldsymbol{\phi}_k}$. Furthermore, the parameter $\boldsymbol{\phi}^k$ of the most likely action gives us the parameterization of the recognized movement, e.g., the pointed at position $\boldsymbol{p}^{uv}$ in the table-top scenario, which is given by the bilinear interpolation parameters $(u, v) = \boldsymbol{\phi}^k$.

In our table-top experiments there are up to nine exemplar HMMs in the PHMM. Therefore, the estimate of the parameter $\boldsymbol{\phi}$ is done in a hierarchical way. In a first step $\boldsymbol{\phi}$ is estimated based on the PHMM given by bilinear interpolation of the outermost four exemplar positions or HMMs. In a second step $\boldsymbol{\phi}$ is refined as an estimate using four exemplar HMMs, which are nearest to the previous estimate.

5 Experiments

In our experiments we focus our considerations on pointing and grasping actions, which are in common action scenarios probably two of the most important movements. The pointing movements are movements such as "This object there..." (Fig. 1). Our grasping movements are reaching towards a particular object in order to grasp it.

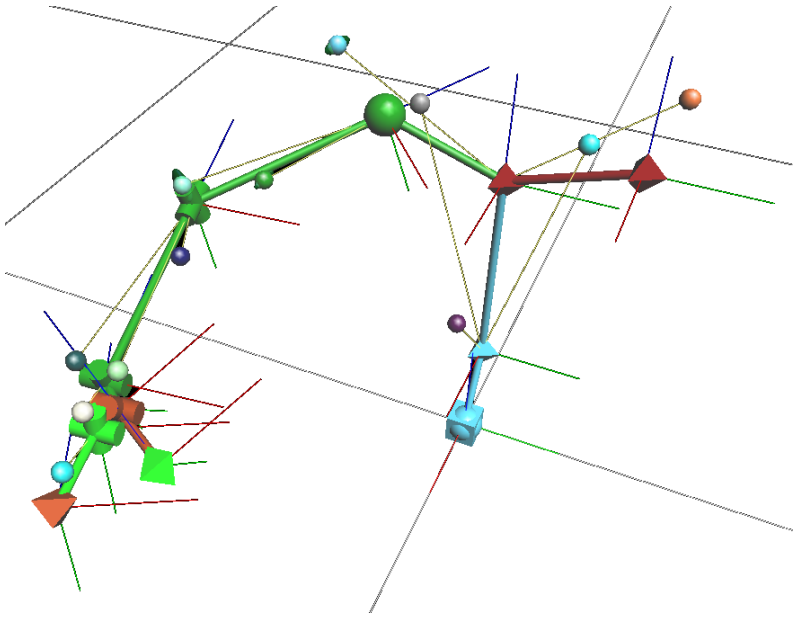

Figure 5: Capture Model of Right Arm. This model is used for motion capturing, for what the model's markers (tiny balls in picture) are aligned to captured marker positions (compare to Fig. 1).

In our systematic experiments we limit our considerations on extensive data that is acquired using a motion capture system. This way, we exclude the vision problem and are able to focus only on the representational issues for movement representation. Based on this data, we evaluate the synthesis and recognition performance of our PHMM approach. In addition, we compare the results to the results that are yielded by that type of PHMMs, which has been proposed by Wilson and Bobick [12]. However, we consider only the linear case of their model. — Concerning online recognition and synthesis, we have first results in a form of an online video, but our experiments based on visual stereo tracking data are still ongoing.

The motion capture data of our systematic experiments is acquired with an eight camera visual marker motion capture system of *Vicon*. For capturing, a model of the right arm (see Fig. 5) is aligned to visual-captured marker positions. The recognition and synthesis experiments are based on seven 3D points located at different segments of the model's body. Capturing speed is 60Hz. The seven data points are located at: sternum; shoulder, and elbow of the right arm; knuckles, index finger, and thumb of the right hand.

The setup of the capture session for acquiring takes place, as follows: The person or actor sits in front of a table (see Fig. 1). The actions are performed at a specific table-top position in such a way, that it is starting and ending in a base position (arm hanging down).

The exemplar positions at table-top form a regular raster, which covers a region of 80cm $\times$ 30cm (width $\times$ depth). For training, a 3×3 raster is used, where 10 repetitions have been recorded for each exemplar position and each action type (pointing, grasping). For evaluation, a 5×7 raster is used, with 4 repetitions for each position to allow a good evaluation statistic (all in all several hundreds of repetitions).

5.1 Training: Setup of PHMMs

The setup of the exemplar HMMs of the PHMMs for the grasping and pointing movements are done as described in Sect. 4.2. Training is done as described in Sect. 4.2. We train the PHMMs based on data of the full 3×3 raster (9 exemplar HMMs) or based on a 2×2 raster, which consists of the four corner exemplar positions of the 3×3 raster. These PHMMs, will be refered in the following as 3×3 or 2×2 PHMM of grasping or pointing. The linear PHMM developed by Wilson and Bobick is also trained

by using the training data of the 3×3 or 2×2 raster, which is refered by us as "Wilson's $n \times n$-trained PHMM".

The PHMMs are setup as follows: The training sequences are rescaled to 100 samples. The PHMMs have 20 states, where the hidden state sequences are forced to stay between 4 and 6 steps in each state.

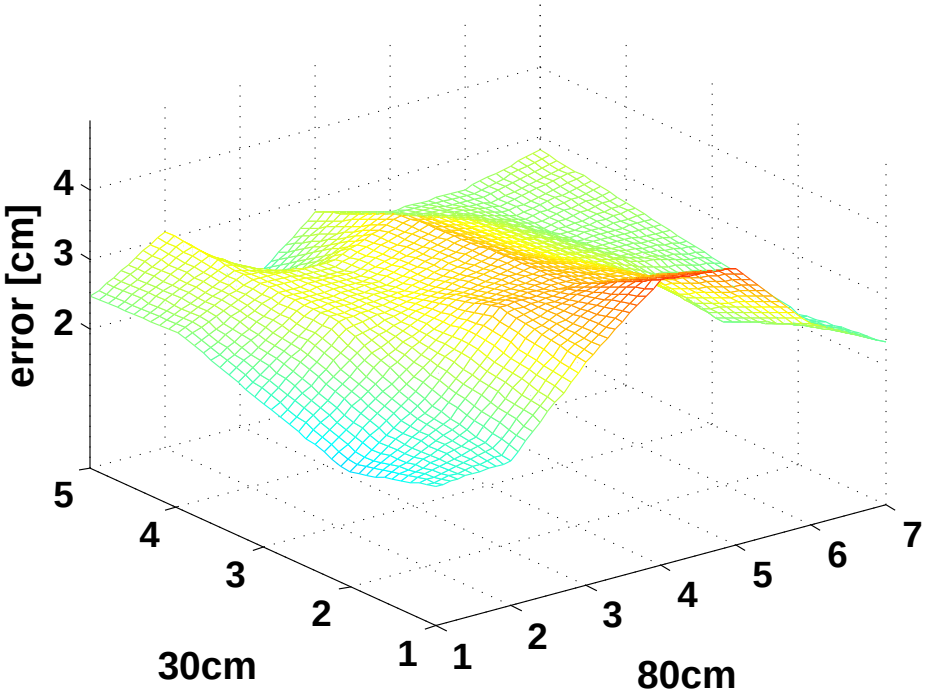

Figure 6: Synthesis Error of Pointing for 2×2 PHMM.

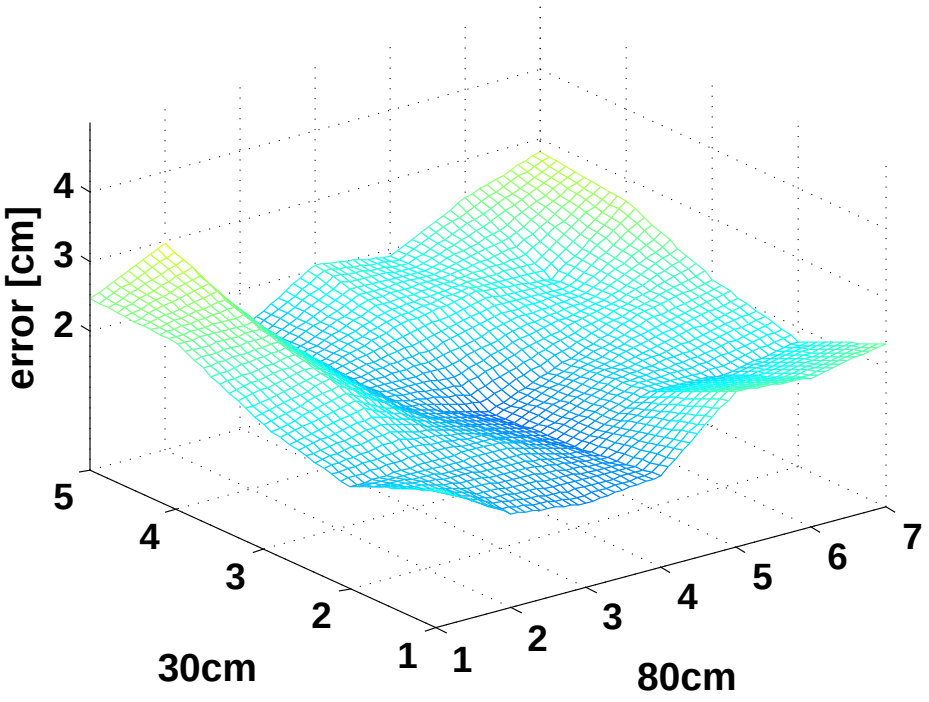

Figure 7: Synthesis Error of Pointing for 3×3 PHMM.

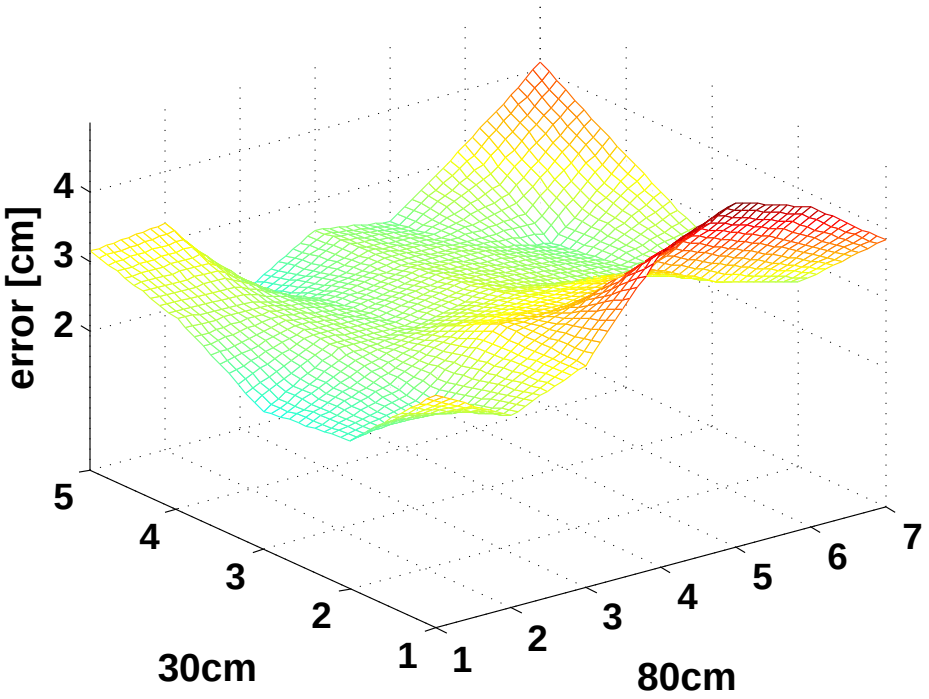

Figure 8: Synthesis Error of Pointing for Wilson's 2×2-trained PHMM.

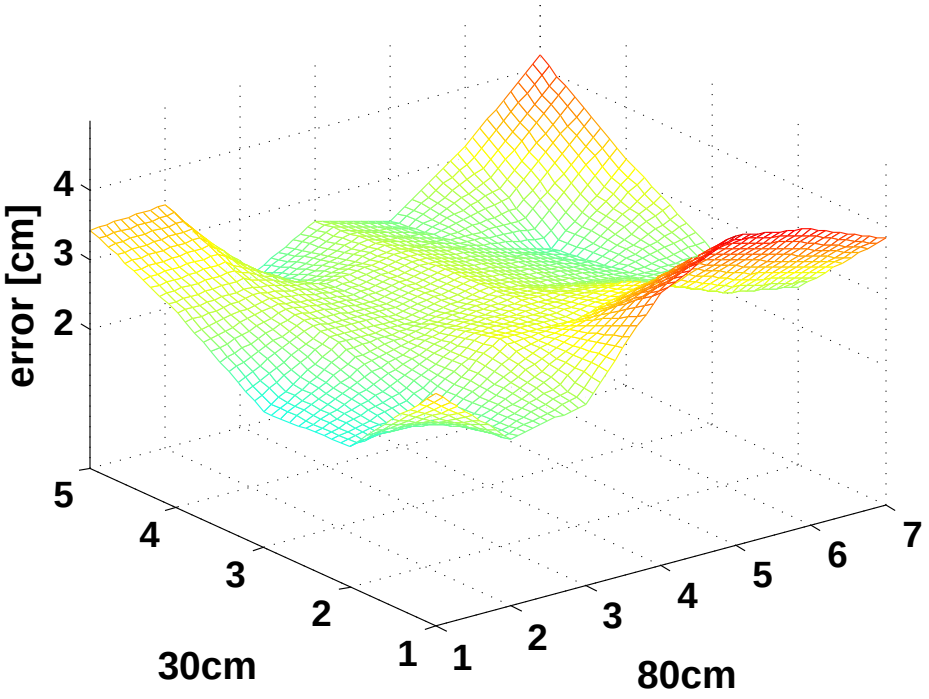

Figure 9: Synthesis Error of Pointing for Wilson's 3×3-trained PHMM.

5.2 Synthesis

Synthesis is done as described above in Sec. 4.1. The performance of synthesis is systematically evaluated by plotting the synthesis error for each of the 5×7 positions, for which test exemplars have been recorded.

The error calculation for each of the 35 synthesized movements for the raster is done as follows: The

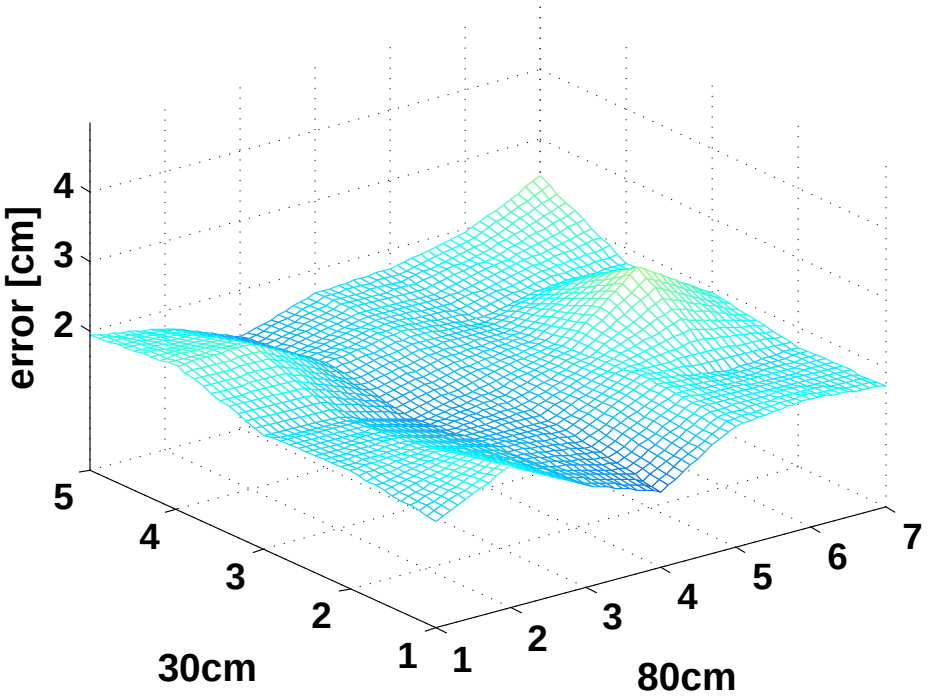

Figure 10: Synthesis Error of Grasping for 3×3 PHMM.

error is calculated as a distance measure between the synthesized movement and a statistical ground truth estimate which is based on the four test exemplars. Therefore, the four test movements of a specific position are averaged by first training an 80 state HMM with the test movements and by then re-synthesizing the average movement $\bar{\boldsymbol{f}}(t) = (\bar{\boldsymbol{f}}_i(i))_{i=1}^{7}$ from the HMM. where the $\boldsymbol{f}_i(t)_{i=1,\ldots,7}$ are 3D trajectories (e.g., of the wrist, elbow...).

The error ε of the synthesized movement $\boldsymbol{f}(t) = (\boldsymbol{f}_i(t))_{i=1}^{7}$, which is synthesized based on the PHMM, is calculated as the route-mean-square error between the time warped synthesis, $\boldsymbol{f}(t)$, and reference, $\bar{\boldsymbol{f}}(t)$:

$$\varepsilon = \sqrt{\int \sum_{i=1}^{7} \frac{(\boldsymbol{f}_i(\alpha(t)) - \bar{\boldsymbol{f}}_i(\bar{\alpha}(t)))^2}{7} \mathrm{d}t \Big/ \int \alpha(t) dt}, \tag{3}$$

where $\alpha(t)$ and $\bar{\alpha}(t)$ are warping functions. The calculation of ε is based on the super-sampled sequences using linear interpolation. As the starting and ending points of the reference $\bar{\boldsymbol{f}}(t)$ do vary slightly, the first and last 10% of the sequences are not considered in the error measure.

The Figs. 6, and 7 compare the synthesis errors for the pointing movement over the 5×7 raster (covering a table-top range of 80cm×30cm) for our PHMM approach based on 2×2 and 3×3 exemplar HMMs. Clearly, the performance in the middle of the covered region increases, if the 3×3 PHMM is used. The Figs. 8, 9 show the performance of Wilson's PHMM. Here, the performance does not change dramatically for a training raster of higher resolution. This is, however, not surprisingly as we use only the linear type of Wilson's PHMM. Fig. 10 shows that the results for the grasping action are very similar to the pointing actions.

The synthesis errors are approximately 1.8cm for our PHMM for grasping and pointing, if the outer regions are neglected, where the pose of the person is extremely stretched. For the linear type of Wilson's PHMM, the errors are slightly higher (≈2.5).

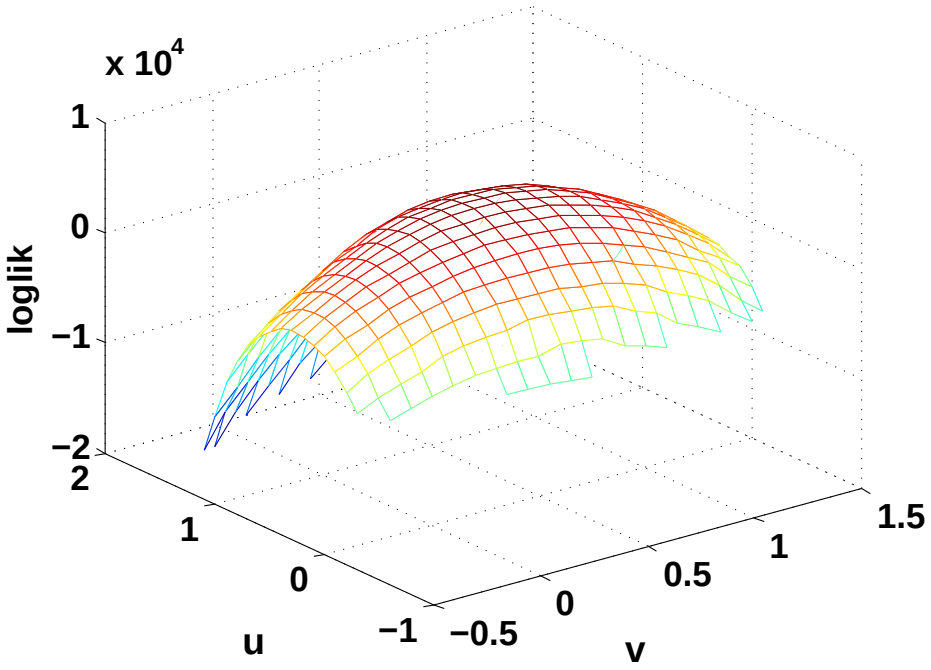

Figure 11: Loglik of the Model Parameters (u, v) given a sequence of Parameterization $(0.5, 0.5)$. The interval $[0, 1]^2 \ni (u, v)$ is mapped to the table-top region of 80cm$\times$30cm.

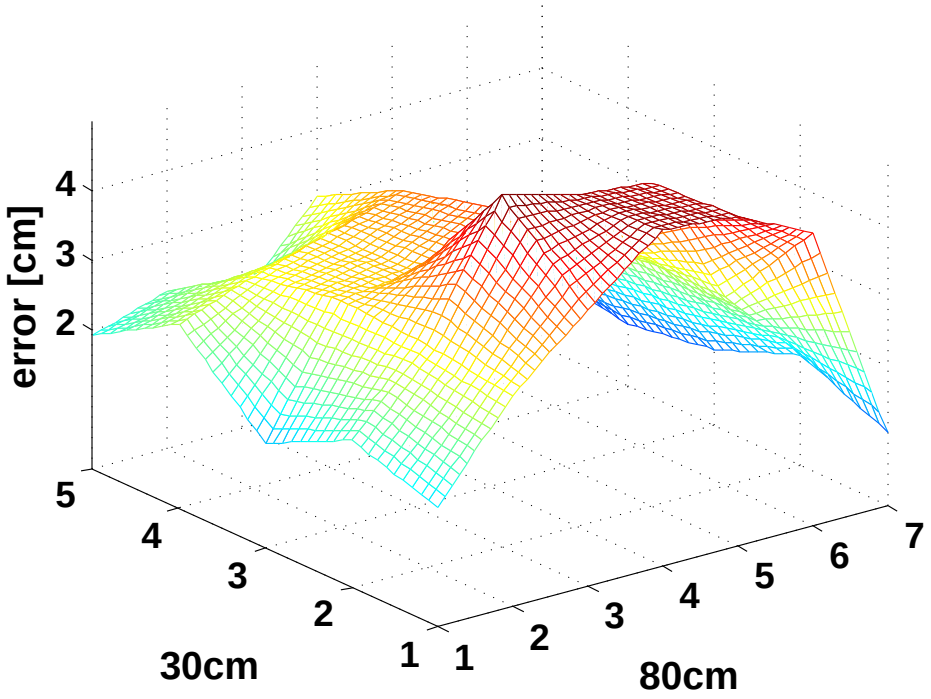

Figure 12: Recognition Error of Pointing for 2$\times$2 PHMM.

5.3 Recognition

Here, two things are to be considered: the recognition performance in terms of the recognized associated position of an action, and rate of correct classifications of the types of the test actions.

In advance, it is worth to take a look at Fig. 11, which gives a hint that the optimization problem of maximizing the log likelihood function $f(u, v) = \log P(\boldsymbol{X}|\boldsymbol{\lambda}^{uv})$ given a movement $\boldsymbol{X}$ is tractable

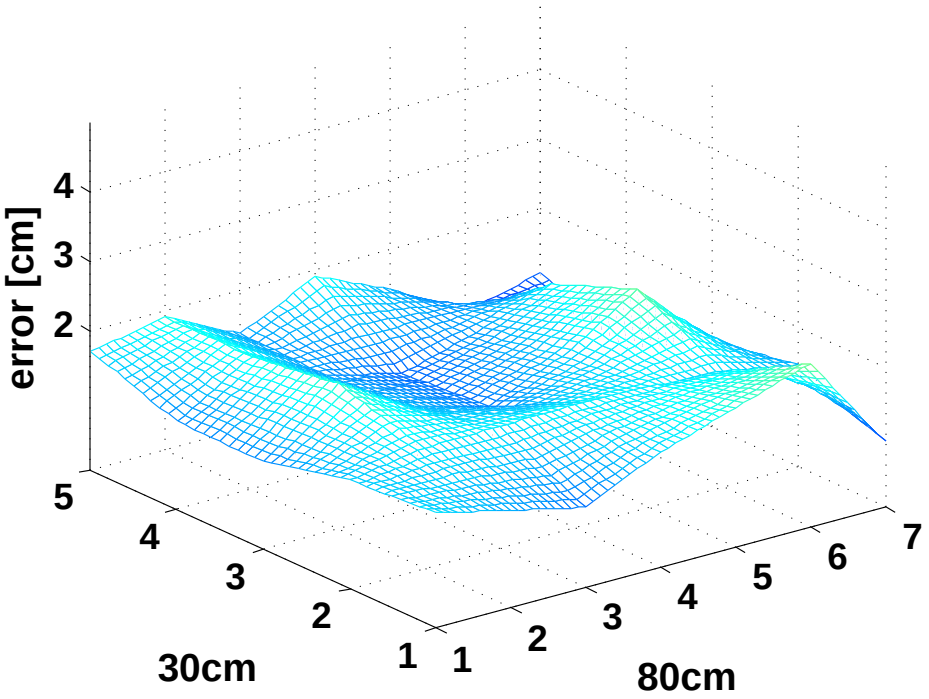

Figure 13: Recognition Error of Pointing for 3×3 PHMM.

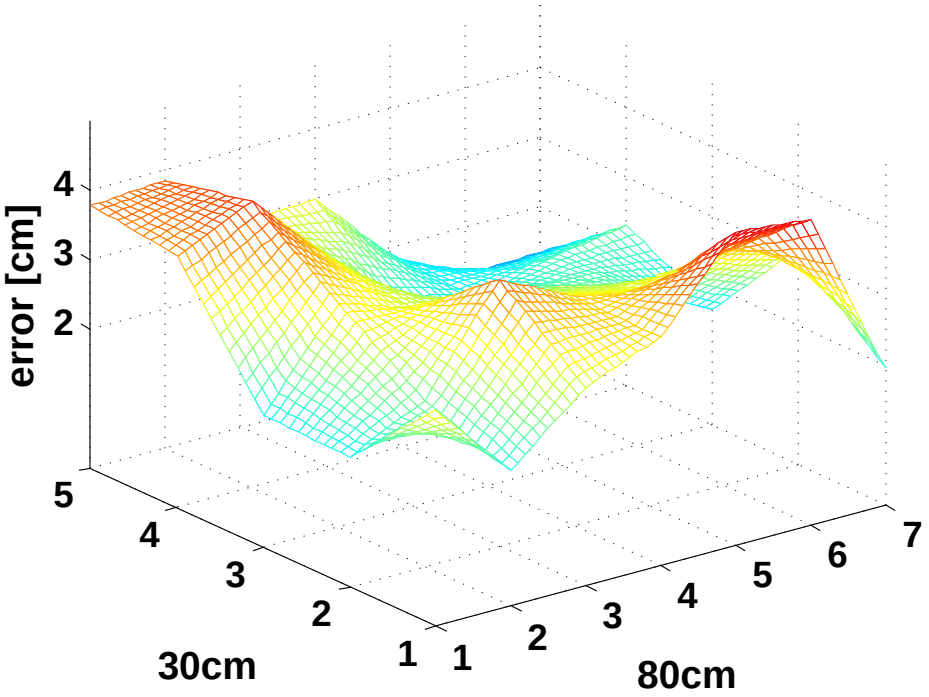

Figure 14: Recognition Error of Pointing for Wilson & Bobick's 3×3-trained PHMM.

by standard optimization techniques (smoothness and strict convexity). In this case, the most likely parameterization (u, v), or associated table-top position of $\boldsymbol{X}$ can be easily estimated. However, in our experiments it has turned out that the maximum of $f(u, v)$ is sometimes a very sharp peak. To address this problem, the function can be smoothed for the first iterations of the optimization by increasing the covariances of the model's Gaussians.

The recognition performance of the associated table-top positions behave very similar to the results

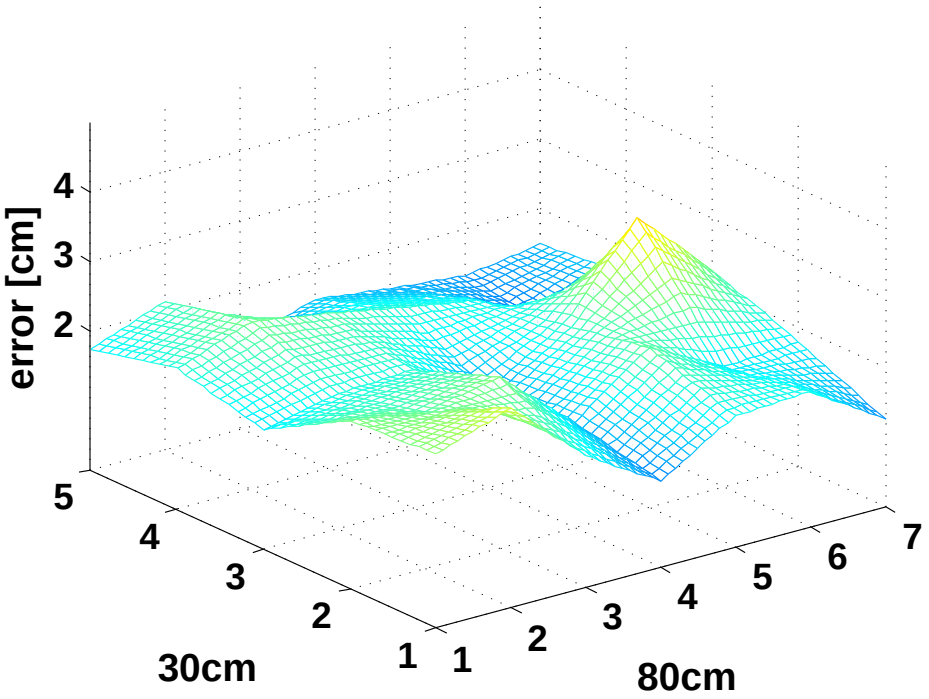

Figure 15: Recognition Error of Grasping for Wilson's 3×3 PHMM.

of synthesis. The error for each position of the 5×7 raster are calculated as the average deviation of the estimated position and the ground truth position for all four test example movements. The recognition performance of our PHMM for pointing and grasping and the performance of Wilson and Bobick's PHMM are presented (Fig. 12–15). Again, clearly, the performance increases in the inner region for our 3×3 PHMM (Fig. 12) compared to the 2×2 PHMM (Fig. 13).

The recognition performance of our 3×3 HMM are similar to Wilson and Bobick's linear type of PHMM (averaged errors of ≈2cm, and slightly smaller for our PHMM).

The rate of right-classified types of the 280 grasping and pointing test movements decreases from 94% to 93% by using the 3×3 PHMMs instead of the 2×2 PHMMs. It is 95% for Wilson and Bobick's PHMM, independently from the used training data (data of the 3×3 or 2×2 raster).

5.4 Online Recognition

Our online demo [REF] shows the applicability of our approach for online recognition of pointing, the position pointed at, and also for motion synthesis. For the synthesis of the robot's arm movement, a PHMM is used, that is trained by the data used in the experiments above. The recognition is based on the position of the elbow and wrist, that are estimated by our online body tracker based on 3D data of a stereo head camera. For the recognition, a PHMM is trained for the last part of pointing movements. The recognition of the position is estimated as the most likely parameter of the PHMM over a recent time window of the elbow and wrist positions, which is recognized as pointing by simple thresholding.

6 Conclusion

We have presented and evaluated a novel approach to handle recognition and synthesis of movements of particular type (sematic), which vary in a parametric way. The basic idea is to incorporate the parameterization of the movements into the HMM (PHMM). Contrary to Wilson and Bobick [12], where the model learns the variation of the Gaussian means, we align some exemplar HMMs for specific pa-

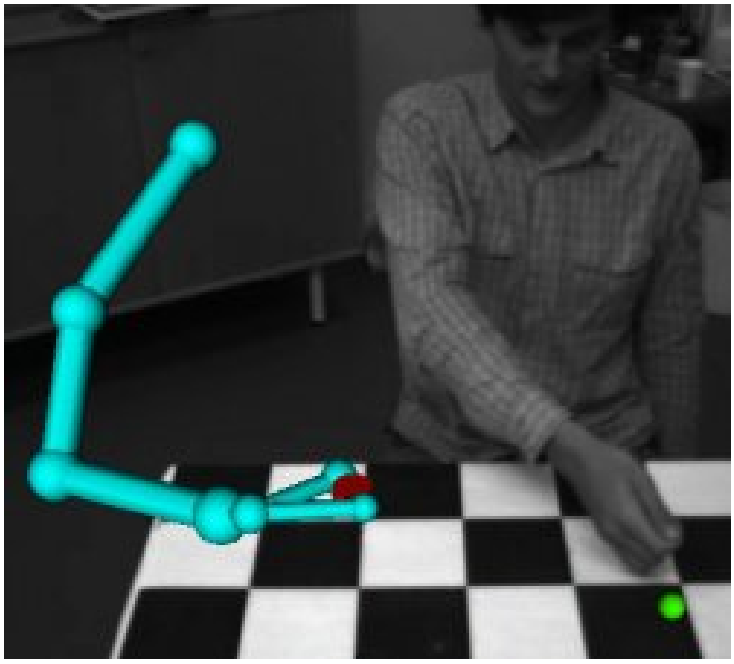

Figure 16: Online Demo. A person is advising a virtual robot arm to relocate a red object (currently, in hand of the robot) at table-top. A person is pointing at the new position. The ball nearby the person's hand indicates the recognized position. The current color (green) of the ball indicates a high likelihood of a pointing movement.

rameterizations, so that the interpolation between the Gaussians is senseful. In our approach all PHMM parameters are allowed to vary depending on the parameterization, unlike in [12].

The experiments show the applicability of our approach for synthesis and recognition of movements (errors ≈2cm), where the performance is similar compared to that of Wilson and Bobick's approach. The classification rate is for both approaches similar $\approx 94\%$.

Finally, it's worth mentioning—even though, we did not compare to the nonlinear case of Wilson and Bobick's approach—that our approach should perform better in such cases, where the movements do vary strongly (as all PHMM parameters can change), with the draw back that several exemplar HMMs have to be setup.

Acknowledgment
This work was partially supported by EU through grant PACO-PLUS, FP6-2004-IST-4-27657.

References

[1] T. Asfour, K. Welke, A. Ude, P. Azad, J. Hoeft, and R. Dillmann. Perceiving objects and movemetns to generate actions on a humanoid robot. In *Proc. Workshop: From features to actions – Unifying perspectives in compnational and robot vision, ICRA*, Rome, Italy, April 2007. 1, 3

[2] B. Dariush. Human Motion Analysis for Biomechanics and Biomedicine. *Machine Vision and Applications*, 14:202–205, 2003. 1

[3] S. Gunter and H. Bunke. Optimizing the number of states, training iterations and gaussians in an hmm-based handwritten word recognizer. *icdar*, 01:472, 2003. 3

[4] X. Huang, Y. Ariki, and M. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990. 3

[5] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991. 3, 4

[6] C. Lu and N. Ferrier. Repetitive Motion Analysis: Segmentation and Event Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):258–263, 2004. 3

[7] T. Moeslund, A. Hilton, and V. Krueger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–127, 2006. 3

[8] D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie. Learning and Tracking Cyclic Human Motion. In *Workshop on Human Modeling, Analysis and Synthesis at CVPR*, Hilton Head Island, South Carolina, June 13-15 2000. 1

[9] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986. 3

[10] S. Schaal. Is Imitation Learning the Route to Humanoid Robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999. 1

[11] D. Vecchio, R. Murray, and P. Perona. Decomposition of Human Motion into Dynamics-based Primitives with Application to Drawing Tasks. *Automatica*, 39(12):2085–2098, 2003. 3

[12] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):884–900, 1999. 2, 3, 4, 7, 13, 14